



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/649,903	08/26/2003	Mahesh A. Ramchandani	5150-77400	5752
7590	06/10/2011		EXAMINER	
Jeffrey C. Hood Meyertons, Hood, Kivlin, Kowert & Goetzel PC P.O. Box 398 Austin, TX 78767			MITCHELL, JASON D	
			ART UNIT	PAPER NUMBER
			2193	
			MAIL DATE	
			06/10/2011	DELIVERY MODE
				PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte MAHESH A. RAMCHANDANI

Appeal 2009-012553
Application 10/649,903
Technology Center 2100

Before KARL D. EASTHOM, DANIEL S. SONG, and GREGORY J. GONSALVES, *Administrative Patent Judges*.

GONSALVES, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134(a) from the rejection of claims 76-86, 88-107, and 109-114. (App. Br. 2.) Claims 1-75, 87, and 108 were cancelled. (*Id.*) We have jurisdiction under 35 U.S.C. § 6(b). We affirm.

The Disclosed Invention

The disclosed invention includes “a system and method for creating a test executive application that provides a graphical user interface for executing one or more test executive sequences.” (Spec. 3:1-3.) A “graphical user interface (GUI) element may be included in the test executive application in response to user input.” (Spec. 3:9-10.) A “control may also be included in the test executive application in response to user input.” (Spec. 3:17-18.) The control may include “functionality for managing execution of a test executive sequence and/or displaying information regarding execution of a test executive sequence.” (Spec. 3:21-23.) A binding may be configured between the GUI element and the control so that the control performs its functionality automatically in response to user input received at the GUI element. (Spec. 3:25-28.)

Exemplary claim 76 follows:

76. A computer-implemented method for displaying information regarding a test executive sequence, wherein the test executive sequence includes a plurality of steps, the method comprising:

including a GUI element in a graphical user interface of a run-time operator interface application in response to user input, wherein the GUI element is operable to display information;

including a control in the run-time operator interface application in response to user input, wherein the control includes pre-existing first functionality for determining the steps in the test executive sequence;

configuring a binding between the GUI element and the control, wherein configuring the binding enables the GUI element to automatically display at least a subset of the steps in the test executive sequence in response to the control determining the steps in the test executive sequence during execution of the run-time operator interface application; and

executing the run-time operator interface application, wherein said executing comprises the control executing to automatically determine the steps in the test executive sequence, wherein the binding between the GUI element and the control causes the GUI element to automatically display at least a subset of the steps in response to the control determining the steps, wherein the GUI element displays the at least a subset of the steps in the graphical user interface of the run-time operator interface application during execution of the run-time operator interface application.

The Examiner rejected claims 76-86, 88-90, 92-107, and 109-114 as obvious under 35 U.S.C. § 103(a) based on U.S. Patent 6,401,220 B1 (Grey) and U.S. Patent 5,485,617 (Stutz). (Ans. 3.)

The Examiner rejected claim 91 as obvious under 35 U.S.C. § 103(a) based on Grey, Stutz, and U.S. Patent 6,718,534 B1 (Carter). (Ans. 14.)

ISSUE

Appellants' responses to the Examiner's positions present the following issue:

Did the Examiner establish that the combination of Grey and Stutz discloses a) “including a control in the run-time operator interface application in response to user input, wherein the control includes pre-existing first functionality for determining the steps in the test executive sequence,” b) “configuring a binding between the GUI element and the control” to enable the GUI element to automatically display the steps determined by the control, “during execution of the run-time operator interface application” and c) “executing the run-time operator interface application, wherein said executing comprises the control executing to automatically determine the steps” and the binding causes the GUI element to automatically display the steps, as required by claim 76, and as similarly required by claims 94, 95, 96, 113, and 114?

FINDINGS OF FACT (FF)

Grey

1. Grey discloses a “test executive program which provides greatly improved configurability and modularity, thus simplifying the creation, modification and execution of test sequences.” (Abstract.)
2. Grey teaches a test sequence that “comprises a series of steps, wherein a step is typically a test performed on an instrument.” (Col. 4, ll. 47-48.) A step may have a type (*i.e.*, a step type). (Col. 4, ll. 47-50.) A step type includes a “custom set of properties and/or operations associated with a step.” (Col. 4, ll. 65-66.) More particularly,

In a test sequence with a number of steps, in many instances the user will desire a number of steps that have some commonality of functionality and/or properties. A primary purpose of a step type is to define common properties and/or

operations associated with a plurality of steps in a single location, referred to as the step type, thereby eliminating the need for the user to define these common properties and/or operations with each of the respective steps.

(Col. 4, l. 51-59.)

3. Grey discloses a “Test Stand Engine that executes sequences, wherein sequences contain steps.” (Col. 4, ll. 59-62).

Grey also discloses a sequence editor. (Fig. 4.) The main tab of the sequence editor displays the steps in a test sequence. (Fig. 4). The sequence editor executes test sequences: “In the TestStand sequence editor 212, the user can start multiple concurrent executions. Multiple instances of the same sequence can be executed and different sequences can be executed at the same time.” (Col. 12, l. 65- col. 13, l. 1.) “[T]he sequence editor ... interface[s] to the test executive engine.” (Col. 3, ll. 16-18.)

Stutz

4. Stutz discloses a “method and system for dynamically generating object connections.” (Abstract.)

5. In Stutz, “a visual programmer specifies the visual components [and] also the interconnections between various ports.” (Col. 10, ll. 46-49.)

PRINCIPLES OF LAW

The Examiner bears an initial burden of factually supporting an articulated rejection. *In re Oetiker*, 977 F.2d 1443 (Fed. Cir. 1992). Under § 103, ““there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.”” *KSR Int’l*

Co. v. Teleflex, Inc., 550 U.S. 398, 418 (2007) (citation omitted).

ANALYSIS

Appellant asserts that the combination of Grey and Stutz does not teach “the limitation of, ‘including a control in the run-time operator interface application in response to user input, wherein the control includes pre-existing first functionality for determining the steps in the test executive sequence.’” (App. Br. 11.) Appellant argues that the passages of Grey cited by the Examiner do not teach this claim limitation because they pertain to the processing of step types instead of steps. (*Id.* at 11-13.) For example, Appellant argues that the cited passages teach “nothing whatsoever about a control that includes pre-existing first functionality for determining the steps (not the step types!) in the test executive sequence. (*Id.* at 11 (emphasis omitted).)

Appellant’s arguments, however, are not persuasive. Grey teaches a test sequence comprising a sequence of steps. (FF 1 and 2.) A step in the test sequence may have a step type. (FF 2.) A step type includes a set of steps. (FF 2.) Furthermore, a step in a step sequence may be an instance of a step type and therefore, a step in a test sequence may include a series of operations or smaller steps.

Moreover, Grey discloses a TestStand engine as a control that determines the type of a step and therefore, determines the set of operations or smaller steps associated with the step in the test sequence. (FF 4.) The TestStand engine also executes test sequences. (FF 4.) Accordingly, Appellant has not shown that Grey fails to disclose a control in a run-time operator interface that determines the steps in the test executive sequence.

Appellant also asserts that Grey does not disclose “configuring a binding between a GUI element and the control” to enable the GUI element to automatically display the steps determined by the control “during execution of the run time operator interface application.” (App. Br. 14 (emphasis omitted.)) Appellant argues that Grey does not teach anything “whatsoever about any steps of the sequence being displayed during execution of the run-time operator interface application.” (App. Br. 13.) More specifically, Appellant argues that the Grey’s sequence editor does not display steps during execution of the run-time operator interface application. (*Id.* at 14-15.)

Appellant’s arguments, however, are not persuasive. The main tab of the sequence editor displays the steps in a test sequence. (FF 3.) Moreover, contrary to Appellant’s arguments, Grey’s sequence editor does execute test sequences. (FF 3.) Moreover, the sequence editor is configured to interface or to be bound to the test executive engine. (FF 3.) Moreover, Stutz teaches that GUI elements and controls can be bound by a user. (FF 5 and 6.) Accordingly, Appellant has not shown that the combination of Grey and Stutz fails to disclose a binding between a GUI element and control to enable the GUI element to display the steps of a test sequence during execution of the run time operator interface application.

Appellant asserts that Grey does not disclose the limitation of “executing the run-time operator interface application, wherein said executing comprises the control executing to automatically determine the steps.” (App. Br. 15-16.) To support this assertion, Appellant presents the same argument as set forth for the claim limitations discussed above. As explained *supra*, these arguments are not persuasive.

Appellant also argues that the Examiner has not established a proper motivation to combine Stutz with Grey because Stutz operates at a different level of programming than Grey. (App. Br. 16.) As explained by the Examiner, however, Stutz discloses the binding of a GUI element with a control and therefore, one of ordinary skill in the art would have been motivated to use the binding disclosed in Stutz in the system of Grey while it was being developed. (Ans. 23; *accord* FF 5 and 6.)

For these reasons, Appellant has not shown that the combination of Grey and Stutz fails to render obvious independent claim 76. Accordingly, we will sustain the Examiner's rejection of independent claim 76. For the reasons expressed above as well as the reasons set forth in the Examiner's answer, we will also sustain the Examiner's rejection of independent claims 94, 95, 96, 113, and 114 and dependent claims 77-86, 88-93, 97-107, and 109-112 because Appellant either did not present separate arguments for those claims or presented arguments that are similar to those set forth for claim 76. (*See* App. Br. 19-29; Reply Br. 3-4; Ans. 5-15, 25-29.)

DECISION

We affirm the Examiner's decision rejecting claims 76-86, 88-107, and 109-114.

No time for taking any action connected with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

Appeal 2009-012553
Application 10/649,903

ELD